

(9) CLAIMS

What is claimed is:

1 1. A method for cycling trigger-event operations
2 of a networked computer peripheral, the method comprising:
3 periodically monitoring at least one current
4 timestamp service across a network connection;
5 recording a first timestamp indicative of
6 time of current trigger-event;
7 using the first timestamp and current
8 timestamp, calculating elapsed time since a last trigger-
9 event; and
10 re-running said trigger-event operations only
11 when the elapsed time exceeds a pre-specified period for
12 cycling operations.

1 2. The method as set forth in claim 1, said
2 calculating further comprising:
3 including accounting for any peripheral
4 power-off time period.

1 3. The method as set forth in claim 1, said
2 periodically monitoring further comprising:
3 monitoring coordinated universal time from a
4 network time protocol server.

1 4. The method as set forth in claim 2,
2 comprising:
3 when no power-off condition occurs during
4 operation of the peripheral, calculating elapsed time, ET,
5 since a previous trigger-event operation in accordance with
6 an equation:

$$ET = CT_R - TOTE,$$

8 where CT_R is the current real time and TOTE is a recorded
9 timestamp indicative of the trigger-event, and a
10 determination:

$$\text{is } ET > P_{MAX} \text{ ?},$$

12 where " P_{MAX} " is a maximum period of operation pre-specified
13 before re- is to be run.

1 5. The method as set forth in claim 2,
2 comprising:
3 when a power off condition occurs during
4 operation of the peripheral, calculating elapsed time since
5 a previous trigger-event operation in accordance with an
6 equation:

$$ET = RT - TOS,$$

8 where TOS is the approximate time of power off condition,
9 and

10 RT is a timestamp indicative of a following power on
11 condition, and a determination:

$$\text{is } ET > P_{MAX} \text{ ?}.$$

10 timestamp, computer code calculating elapsed time since a
11 last trigger-event cycle; and
12 computer code re-running said calibration
13 operations only when the elapsed time exceeds a pre-
14 specified period for cycling operations.

1 10. The memory as set forth in claim 9, said
2 computer code calculating further comprising:
3 including accounting for any peripheral
4 power-off time period.

1 11. The memory as set forth in claim 9, said
2 computer code periodically monitoring further comprising:
3 monitoring coordinated universal time from a
4 network time protocol server.

1 12. The memory as set forth in claim 10,
2 comprising:
3 when no power-off condition occurs during
4 operation of the peripheral, computer code calculating
5 elapsed time, ET, since a previous calibration operation in
6 accordance with an equation:

$$ET = CT_R - TOTE,$$

8 where CT_R is the current real time and TOTE is a recorded
9 timestamp indicative of the trigger-event, and a
10 determination:

11 is $ET > P_{MAX}$?,
12 where " P_{MAX} " is a maximum period of operation pre-specified
13 before recalibration is to be run.

1 13. The memory as set forth in claim 10,
2 comprising:
3 when a power off condition occurs during
4 operation of the peripheral, computer code calculating
5 elapsed time since a previous calibration operation in
6 accordance with the equations:

7 $ET = RT - TOS$,
8 where TOS is the approximate time of power off condition,
9 RT is a timestamp indicative of a following power on
10 condition, and

11 is $ET > P_{MAX}$?.

1 14. The memory as set forth in claim 9, the code
2 for cycling calibration operations of a networked computer
3 peripheral comprising:
4 a Java application on a Java Virtual Machine.